Medição da Intenção de Voto por Meio de Análise de Sentimentos em Tweets.

Caíque Leonardo Freitas Tosta[1], **Rafael Geraldeli Rossi**[1]

[1] Universidade Federal do Mato Grosso do Sul (UFMS)
UNID. II: Av. Ranulpho Marques Leal, 3484 – CEP 79620-080 – Cx Postal nº210
Mato Grosso do Sul – MS – Brazil

caique.tosta@hotmail.com.br, rafael.g.rossi@ufms.br

Abstract. *Vote intentions surveys are common to be conducted close to election periods. Their results have an impact on both the decision-making of political parties and electors about which candidate to vote. These surveys are usually conducted manually through interviews and are limited to a specific region with a number of respondents who are often far from the reality of the electoral universe. This paper presents the use of machine learning techniques for automatic classification of sentiment in tweets and uses them to measure the voting intention of the mayor municipal election of São Paulo city in 2016. We present that the machine learning approaches had satisfactory results in the automatic classification of sentiment in tweets and through the measurement of intention to vote it was possible to approach results of the first surveys of intention to vote and rejection rate, however, neither the vote intentions surveys nor the approach used in this paper were able to accurately approximate with the real election results.*

Resumo. *As pesquisas de intenção de voto são comuns de serem realizadas próximas a períodos eleitorais e seus resultados têm impacto na tomada de decisão tanto de partidos políticos quanto de eleitores sobre qual candidato escolher. Essas pesquisas costumam ser realizadas de forma manual através de entrevistas e são limitadas a uma região específica com um número de entrevistados que costuma estar longe da realidade do universo eleitoral. Este trabalho apresenta o uso de técnicas de aprendizado de máquina para classificação automática de sentimentos em tweets e as utiliza para medir a intenção de voto da eleição municipal para prefeito da cidade de São Paulo de 2016. Apresentamos que as abordagens de aprendizado de máquina tiveram resultados satisfatórios na classificação automática de sentimentos em tweets e através da medição de intenção de voto foi possível se aproximar de resultados das primeiras pesquisas de intenção de voto e taxa de rejeição, porém nem as pesquisas de intenção de voto e a abordagem usada nesse trabalho conseguiram se aproximar com exatidão do resultado real da eleição.*

1. Introdução

As pesquisas de intenção de voto são realizadas pelos institutos de pesquisas de opinião pública tais como o Ibope, Datafolha e Vox Populi. As pesquisas relacionadas a intenção de voto são muito comuns de serem realizadas próximo a períodos eleitorais, que formam

base para incentivar campanhas políticas e têm impacto quanto na tomada de decisão por parte dos comitês de campanha [REIS 2003] quanto pelos eleitores sobre qual candidato escolher [Ferraz et al. 1996].

Geralmente as pesquisas de intenção de voto são realizadas de forma manual, através de entrevistas com algumas centenas de pessoas, onde o número de pessoas e forma de entrevista são diferentes para cada instituto. O número de pessoas entrevistadas costuma estar longe da realidade do universo eleitoral do Brasil, mas, através de cálculos estatísticos, garantem que são suficientes para representar todos os eleitores. Vale ressaltar que os resultados de pesquisas eleitorais devem ser analisados com cautela durante as campanhas brasileiras, pois podem dar lugar a análises e decisões tomadas sobre uma descrição equivocada da realidade [Gramacho 2010].

Nas redes sociais temos um grande número de usuários, porém como visto no trabalho de [Ceron et al. 2014], apesar dos usuários da internet não necessariamente representarem toda uma população, a análise mostra que, com algumas exceções, têm a habilidade de prever resultados de eleições políticas. Diariamente é produzido uma grande quantidade de dados textuais que muitas vezes representam as opiniões de usuários sobre um determinado político ou candidato a um cargo político, principalmente postagens em redes sociais como o Twitter[1], que é conhecido por ser uma rede social onde os usuários opinam sobre assuntos diversos utilizando um limite de 140 caracteres. No campo da política, o Twitter se mostra um espaço que potencializa tanto a formação de redes de eleitores quanto para discussões e intrigas políticas [Rossetto et al. 2013]. Porém, devido a grande quantidade de *tweets* que é produzida, é impossível uma análise manual para determinar se os *tweets* são opinativos ou não-opinativos e se os *tweets* opinativos falam bem (polaridade positiva), falam mal (polaridade negativa) ou são neutros com relação aos candidatos. Para facilitar essas análises, técnicas computacionais para auxiliar a realização de tais tarefas são necessárias. Com isso, o objetivo desse artigo é explorar técnicas de Mineração de Textos para automatizar a classificação dos *tweets* em opinativos e não opinativos e posteriormente uma classificação com as polaridades dos textos opinativos em positivos, negativos e neutros e com isso quantificar a opinião da população (que realizam postagens em redes sociais) e verificar se essa opinião é condizentes com os resultados obtidos por institutos de pesquisa ou mesmo pelos resultados reais das eleições.

O restante desse artigo está dividido da seguinte forma. Na Seção 2 são apresentados os trabalhos relacionados. Na Seção 3 é descrito o projeto: Medição da Intenção de Voto por Meio de Análise de Sentimentos em Tweets. Na Seção 4 é descrito uma análise experimental. Na Seção 5 é apresentado as considerações finais e trabalhos futuros.

2. Trabalhos Relacionados

Dos trabalhos que utilizam dados do Twitter para análise de sentimentos, a maioria utiliza aprendizado de máquina para fazer uma classificação automática de textos de forma a identificar polaridades em *tweets*. Em geral, os trabalhos têm um objetivo similar de apresentar uma fonte alternativa para os dados das pesquisas eleitorais [Wang et al. 2012, Ceron et al. 2014, O'Connor et al. 2010].

No trabalho realizado por [Wang et al. 2012], foi desenvolvido um sistema para análise em tempo real de sentimentos da população a respeito dos candidatos à presidência

[1] `http://twitter.com`

dos Estados Unidos no ano de 2012. O algoritmo de classificação usado para a análise de sentimentos foi o Naïve Bayes. O classificador apresentou 59% de precisão nas quatro categorias de classificação: negativo, positivo, neutro e incerto.

No trabalho de [Ceron et al. 2014], foram realizadas análises de três casos: (i) popularidade de políticos italianos ao longo de 2011, (ii) intenção de voto dos usuários da internet na eleição presidencial de 2012 na França, (iii) intenção de voto dos usuários da internet na eleição do legislativo na França. Foi utilizado o método de aprendizado de máquina proposto por [Hopkins and King 2010]. Os resultados mostram que os usuários da internet muitas vezes não representam toda a população, porém, têm a habilidade de prever resultados eleitorais. Além disso, foi obtido uma correlação entre os termos utilizados na mídia social com os termos encontrados nas pesquisas de opinião pública.

No trabalho de [O'Connor et al. 2010], foi realizada uma análise de textos onde optaram por usar uma abordagem baseada em conhecimento linguístico, fazendo uma contagem de instâncias dos termos que têm um sentimento positivo e um sentimento negativo. Foi analisado questionários sobre confiança do consumidor e questionários sobre opinião politica no período de 2008 a 2009 e descobriram que esses questionários têm correlação com a frequência de palavras de sentimento em mensagens contemporâneas do Twitter. Os resultados variam dependendo da base de dados, mas em alguns casos tiveram uma correlação de 80% entre os termos encontrados nos questionários e as frequências de palavras no Twitter com objetivo de mostrar o potencial de fontes de texto como um substituto e suplemento para pesquisas tradicionais.

No trabalho de [Bakliwal et al. 2013], foram realizados três experimentos com classificação de sentimentos em uma base com 2624 *tweets* produzidos durante a eleição geral da Irlanda em fevereiro de 2011. Mesmo omitindo *tweets* que foram rotulados como sarcásticos da seleção, a precisão mais alta obtida foi de 61,6% usando aprendizado supervisionado e scores léxicos baseados em subjetividade.

3. Projeto: Medição da Intenção de Voto por Meio de Análise de Sentimentos em Tweets

Para realizar a medição da intenção de voto por meio da análise de sentimentos em *tweets*, foi escolhido como cenário para a execução deste trabalho as eleições municipais para prefeito da cidade, de São Paulo no ano de 2016. Inicialmente foi realizada uma pesquisa em sites de notícia e pesquisas do Ibope com objetivo de identificar o nome dos principais candidatos dos principais partidos políticos. Posteriormente foi realizado uma busca pelo nome dos candidatos através da ferramenta de busca disponível na própria página do Twitter, para verificar variações dos nomes dos candidatos que eram utilizados nos *tweets* bem com *hashtags* relacionadas aos candidatos. Os nomes e *hashtags* que foram selecionadas como *strings* de busca para este trabalho são apresentadas na Tabela 1.

Tabela 1. *Strings* de busca utilizadas na coleta dos *tweets* por candidato.

Celso Russomanno	Marta Suplicy	Luiza Erundina	Fernando Haddad	João Dória
celso russomano	marta suplicy	luiza erundina	fernando haddad	joao doria
#russomano	#ForaMarta	#erundina	#haddad	#joaodoria
#celsorussomano	#martasuplicy	#luizaerundina	#fernandohaddad	

Após definidas as *strings* de busca, os passos seguintes do projeto foram divididos em três etapas. Na primeira etapa é realizado o processo de coleta dos *tweets* para gerar uma base com *tweets* não rotulados. Na segunda etapa, é realizado o processo de aprendizado de máquina, onde inicialmente os *tweets* passam por uma fase de rotulação e pré-processamento, são estruturados e é aplicado as técnicas de aprendizado de máquina para classificação automática. Na terceira etapa é realizado uma extração das estatísticas a partir dos rótulos atribuídos automaticamente aos *tweets* por meio da aplicação de técnicas de aprendizado de máquina. Essas etapas, bem como as interações entre elas são apresentadas na Figura 1. Nas subseções seguintes são apresentados os dados sobre a eleição municipal de São Paulo de 2016 e a descrição de cada uma das três etapas deste projeto.

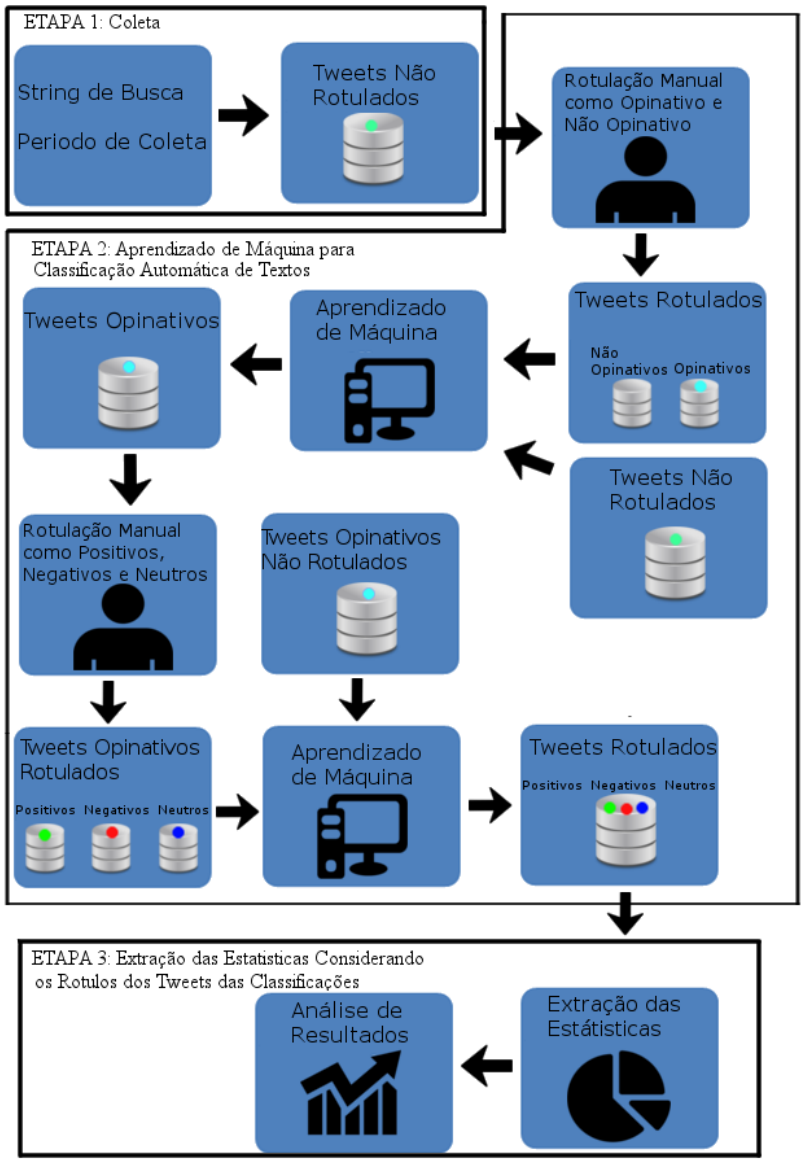

Figura 1. Etapas no desenvolvimento deste projeto.

3.1. A Eleição Municipal de São Paulo de 2016

A eleição municipal da cidade de São Paulo aconteceu no dia 2 de outubro de 2016. Através de pesquisas em sites de notícias e em pesquisas do Ibope foram identificados

os principais candidatos dos principais partidos políticos, sendo os seguintes: João Doria pelo PSDB, Fernando Haddad pelo PT, Celso Russomanno pelo PRB, Marta Suplicy pelo PMDB e Luiza Eundina pelo PSOL. Com 5.789.891 votos validos a eleição foi decidida no primeiro turno com 53,29% dos votos para o candidato João Doria.

3.2. Coleta

O acesso aos dados do Twitter foi realizado usando a Search API[1] que é parte das REST APIs[2], um conjunto de APIs que dá acesso aos dados do Twitter. Com a Search API é possível buscar por uma amostra de *tweets* recentes publicados nos últimos sete dias.

Para o desenvolvimento da ferramenta foi escolhida a linguagem Java, pois através de pesquisas foi identificado a biblioteca Twitter4J[3] que interage com a Search API, disponibilizando funções que interpretam o JSON recebido pela API e as estruturas em componentes JavaBean[4], estruturando os *tweets* em objetos contendo atributos como conteúdo, autor e data. A ferramenta é apresentada na Figura 2. Para utilizar a ferramenta é necessário informar uma *string* de busca (pode ser informado mais de uma, separando as por ";"), a quantidade de interações (nº consulta para coleta de *tweets* na API), intervalo entre cada interação e o diretório para salvar os *tweets*. Outra opção da ferramenta é a análise de termos que pode ser acessada no menu da ferramenta, ao ser selecionada essa opção é realizada uma contagem dos termos encontrados nos textos em um diretório selecionado da tela principal. Para isso, é exibida uma tabela contendo os termos e as *hashtags* encontradas nos *tweets* que são ordenadas decrescentemente pela sua frequência, como apresentado na Figura 3. O intuito dessa tabela é apresentar termos e *hashtags* com potencial para serem incluídas nas próximas consultas para aumentar o número de *tweets* coletados.

O período de coleta se iniciou no dia 15 de Agosto de 2016 e terminou no dia 30 de Setembro de 2016. Vale ressaltar que várias interrupções na coleta devido a problemas de queda de energia na UFMS, que resultou em uma perda considerável de *tweets* durante o decorrer da coleta. A quantidade de *tweets* coletadas foi de 17403. O número de entrevistados em pesquisas de intenção de voto do Ibope e DataFolha no período de 21 de Junho de 2016 a 22 de Setembro de 2016 sobre a eleição municipal de São Paulo foi de 8180. Na Tabela 2 é apresentado o número de *tweets* coletados por candidato.

Tabela 2. Número de *tweets* por candidato.

Candidato	**nº de *tweets***
Celso Russomanno	4.725
Marta Suplicy	3.554
Luiza Erundina	902
Fernando Haddad	7.048
João Dória	1.174

[1] `https://dev.twitter.com/rest/public/search`

[2] `https://dev.twitter.com/rest/public`

[3] `http://twitter4j.org/en/index.html`

[4] Componentes de software escritos na linguagem de programação Java que contém um conjunto de atributos com nomenclaturas simples para métodos de acesso.

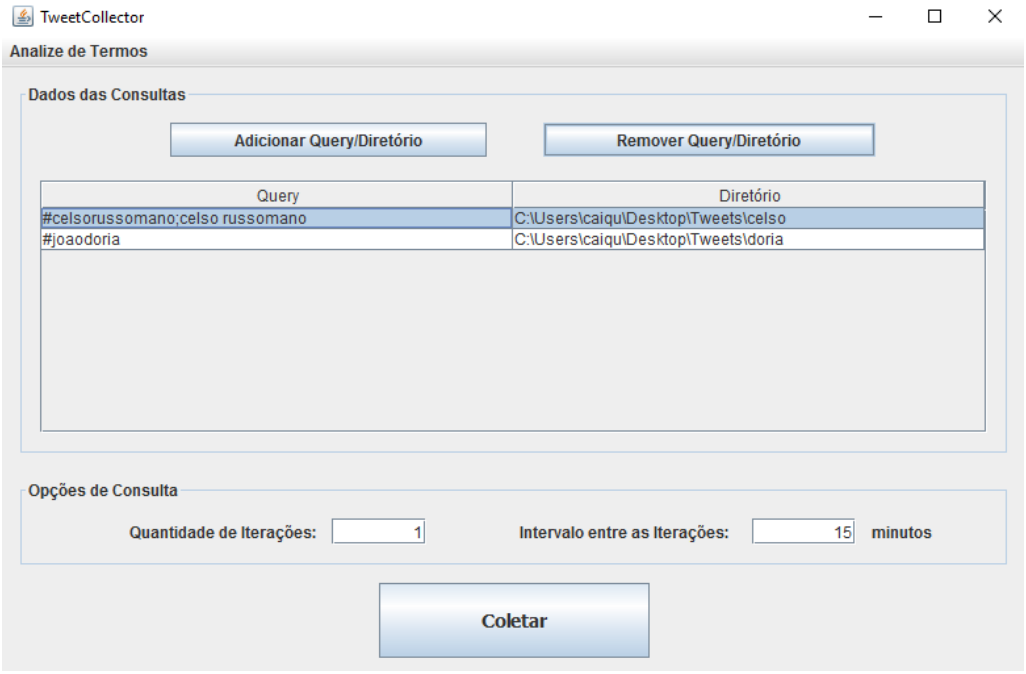

Figura 2. Ferramenta para coleta de *tweets*.

3.3. Aprendizado de Máquina para Classificação Automática de Textos

No processo de aprendizado de máquina, primeiramente é necessário uma etapa de rotulação manual dos textos, essa etapa tem como objetivo gerar uma coleção de textos rotulados com as classes a serem aprendidas pelos algoritmos, com os textos rotulados é necessário representá-los de uma forma pré estruturada para que possam ser interpretados pelos algoritmos. Essa estrutura é utilizada tanto para o processo de aprendizado quanto para avaliar a performance do classificador, assim podendo ser aplicado as técnicas de aprendizado. Nas subseções abaixo são apresentados o processo do aprendizado de máquina.

3.3.1. Rotulação

Para o processo de rotulação foi desenvolvido uma ferramenta na linguagem Java. A primeira tela da ferramenta, apresentada na Figura 4, é destinada para as configurações do processo de rotulação, no qual é necessário informar um diretório onde estão os *tweets* não rotulados e um diretório destino onde serão salvos os *tweets* após serem rotulados. Nesta mesma tela, também devem ser informados os rótulos, que serão utilizados para rotular os documentos. Pode-se também definir a forma como os *tweets* serão apresentados, considerando a ordem ortográfica ou aleatória. Após clicar no botão *Start Labeling* é direcionado a segunda tela, apresentada na Figura 5, a esquerda é apresentado o conteúdo dos *tweets* e a direita são apresentados os rótulos definidos na etapa de rotulação, no qual o usuário pode selecionar um ou mais rótulos. Logo abaixo ficam os botões onde tem a opção para rotular o *tweet* ou pular para o próximo.

A partir dos *tweets* não rotulados obtidos na etapa de coleta, foi realizado uma primeira etapa de rotulação manual. Os *tweets* foram rotulados considerando duas clas-

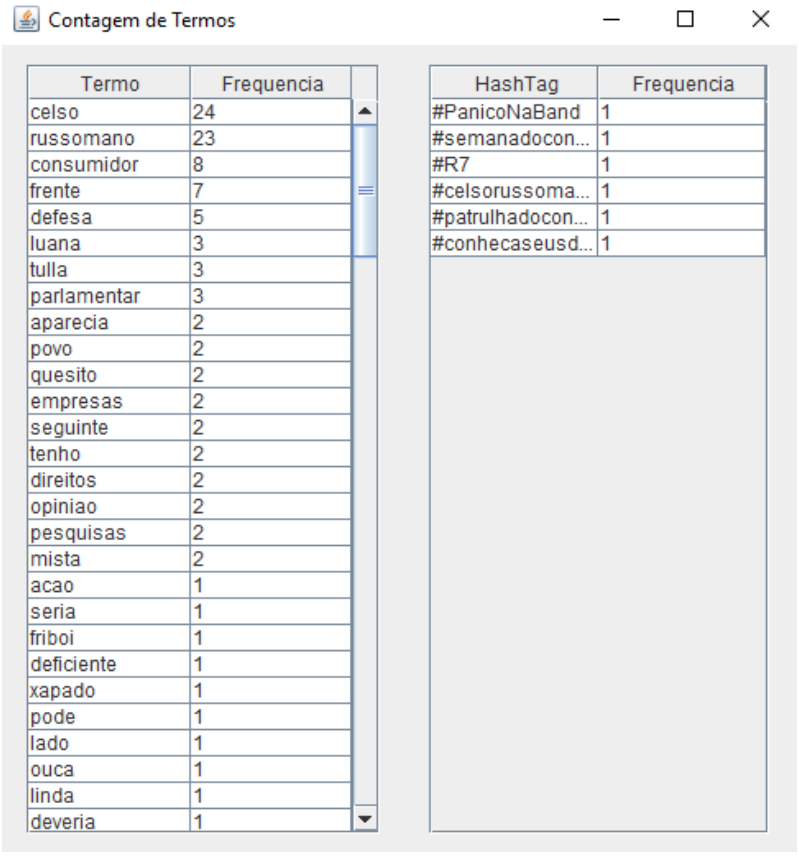

Figura 3. Janela com a contagem dos termos.

ses: opinativos e não opinativos. Exemplos de *tweets* considerados como opinativo e não opinativo são apresentados na Tabela 3. No total foram rotulados 823 *tweets*, sendo 594 opinativos e 229 não opinativos. Considerando esse *tweets*, foi aplicado o processo de aprendizado de máquina para separar os *tweets* não rotulados em opinativos e não opinativos, resultando em um total de 13382 *tweets* opinativos e 3290 *tweets* não opinativos, conforme será melhor detalhado na Seção 3.3.4.

Tabela 3. Exemplo de *tweets* rotulados como opinativo e não opinativo.

Rótulo	***Tweet***
Opinativo	Meu sério, não tem melhor candidato do que o @Haddad_Fernando. Não tem! #DebateDaBand
Não Opinativo	Segundo o PlanejaSampa, @Haddad_Fernando regularizou a situação de 120.383... https://t.co/QqnQKGS56M

Foi realizada uma segunda etapa de rotulação utilizando apenas os *tweets* opinativos. Nessa segunda etapa os *tweets* foram classificados em três classes: positivo, negativo e neutro. Exemplo de *tweets* considerados como positivo, negativo e neutro, são apresentados na Tabela 4. Foram rotulados 337 *tweets* sendo 131 positivos, 101 negativos e 105 neutros. Com esse *tweets* foi aplicado o processo de aprendizado de máquina para separar os *tweets* opinativos em positivos, negativo e neutros, resultando em um total de 4585 *tweets* positivos, 3188 *tweets* negativos e 5272 *tweets* neutros. Novamente maiores

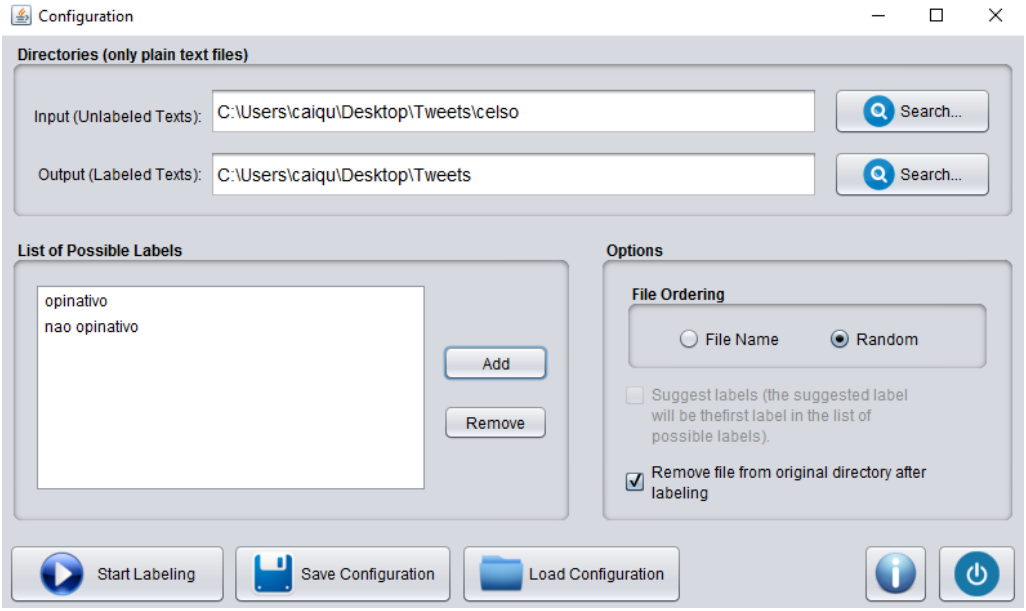

Figura 4. Janela de configuração da ferramenta para rotulação manual de textos.

detalhes sobre o processo serão apresentados na Seção 3.3.4.

Tabela 4. Exemplo de *tweets* rotulados como positivo, negativo e neutro.

Rotulo	***Tweet***
Positivo	Eu vou votar no João Dória pelo menos sei que é um grande administrador. https://t.co/OekAT1OYtU
Negativo	Se o paulistano eleger esse vigarista do Russomanno vai se arrepender mais do que se arrependeu com o Haddad. Celso Russomanno é bandido!
Neutro	A Marta Suplicy é uma mistura de Hebe com o museu de cera

3.3.2. Representação Estruturada Das Coleções

A representação de textos é a base para o processamento de textos e consequentemente para a aplicação de algoritmos de aprendizado de máquina [Rossi 2016]. Porém, antes de gerar as representações estruturadas, alguns processamentos são realizados de forma a eliminar ruídos, palavras desessenciarias e palavras redundantes para o processo de aprendizado de máquina.

Dentre as técnicas de pré-processamento de textos destaca-se a padronização de caixas, remoção de palavras irrelevantes ou ruídos, agrupamento de palavras contendo o mesmo significado em um único atributo, ou ainda seleção de palavras de acordo com uma determinada função sintática. No processo de agrupamento de palavras temos a radicalização que consiste em reduzir as palavras flexionadas para sua base. Por exemplo, as palavras ”gostar”, ”gostando”e ”gostaria”, seriam reduzidas para ”gost”. Um outro processo é a remoção das *stopwords*, que são palavras irrelevantes durante o processo de aprendizado, em geral são artigos, preposições e verbos de estado. Alguns exemplos de *stopwords* são: ”os”, ”as”, ”para”, ”foi”. Após realizar o pré-processamento, deve-se

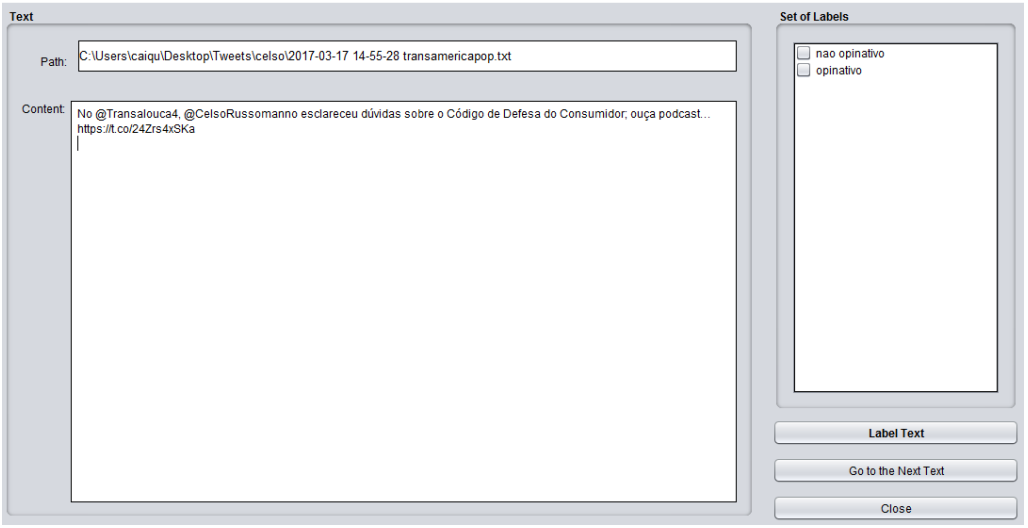

Figura 5. Janela de rotulação da ferramenta para rotulação manual de textos.

escolher um modelo para representar os textos em formato estruturado. O modelo espaço-vetorial tem sido o mais utilizado para a representação de coleções de texto. Basicamente, pode ser visto como uma matriz documento-termo, na qual cada linha representa um documento e cada coluna representa um termo da coleção. Como apresentado na Tabela 5. Normalmente utilizam-se palavras simples como termos da coleção de documentos, gerando a matriz documento-termo denominada *bag-of-words*.

Tabela 5. Matriz documento-termo representando uma coleção com N documentos e M termos.

	$\mathbf{t_1}$	$\mathbf{t_2}$	...	$\mathbf{t_m}$	**Classe**
$\mathbf{d_1}$	w_{d_1,t_1}	w_{d_1,t_2}	...	w_{d_1,t_m}	c_{d_1}
$\mathbf{d_2}$	w_{d_2,t_1}	w_{d_2,t_2}	...	w_{d_2,t_m}	c_{d_2}
...	...	...	...	...	...
$\mathbf{d_n}$	w_{d_n,t_1}	w_{d_n,t_2}	...	w_{d_n,t_m}	c_{d_n}

Outro modelo citado na literatura é a representação de redes. Há várias instanciações desse modelo, mas independe do tipo de rede, todas elas podem ser formalmente definidas como uma tripla $N = (O, R, W)$, na qual O representa o conjunto de objetos da rede, R representa o conjunto das relações entre os objetos e W representa o conjunto de pesos das relações entre os objetos. Um tipo de rede que vem ganhando destaque recentemente e foi utilizada neste trabalho é a rede bipartida. As redes bipartidas são compostas por dois tipos de objetos sendo que um tipo de objeto somente se conecta com objetos do outro tipo. Na Figura 6 é apresentado um exemplo de uma rede heterogênea bipartida utilizada para representar uma coleção com M termos e N documentos[Rossi 2016].

3.3.3. Aprendizado de Máquina

Para o aprendizado de máquina, neste trabalho foram utilizados dois tipos: de aprendizado o Indutivo Supervisionado e Transdutivo Semisupervisionado.

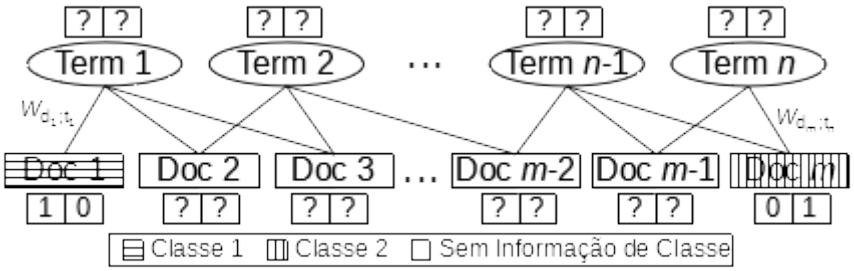

Figura 6. Ilustração de uma rede bipartida para representar coleções de textos.

O aprendizado Indutivo Supervisionado é o mais tradicional, o objetivo da classificação indutiva supervisionada é induzir uma função F que seja capaz de mapear documentos rotulados (D^L) para seus respectivos rótulos (Y^L), isto é, $F : (D^L) \longrightarrow (Y^L)$. A função F induzida por meio do aprendizado indutivo supervisionado (modelo de classificação) é então utilizado para rotular novos documentos [Rossi 2016]. Os algoritmos usados baseados no modelo espaço vetorial são: *Multinominal Naive Bayes* (MNB), *k-Nearest Neightbors* (k-NN), J48, *Support Vector Machines* (SVM). Os algoritmos usados baseados em redes bipartidas são: *Inductive Model based on Bipartite Heterogeneous Networks* (IMBHN^C e IMBHN^R).

O MNB é utilizado para inferir a probabilidade para as classes dos termos. O SVM utiliza apenas documentos rotulados para induzir um hiperplano que melhor separe as classes dos documentos. O J48 é utilizado para induzir uma árvore de decisão, onde os nós folha pertencem a uma mesma classe. O KNN é utilizado para induzir a classe a partir dos vizinhos mais próximos. O IMBHN é utilizado para inferir os pesos dos termos para as classes utilizando a estrutura de rede bipartida [Rossi et al. 2012, Agarwal et al. 2011].

O aprendizado transdutivo semisupervisionado é utilizado quando se tem uma pequena quantidade de exemplos rotulados. Neste tipo de aprendizado os exemplos não rotulados são utilizados para melhorar a performance de classificação em relação a utilização de somente exemplos rotulados. Diferente do aprendizado indutivo, o objetivo do aprendizado transdutivo não é induzir um modelo de classificação para classificar documentos novos, mas classificar diretamente todos os documentos não rotulados já conhecidos. A classificação transdutiva tem como objetivo gerar uma função $F : (O^{L+U}) \longrightarrow (Y^{L+U})$ tal que F seja um bom preditor sobre os dados não rotulados [Rossi 2016]. Os algoritmos usados baseados em redes bipartida são: *Label Propagation through Bipartite Heterogeneous Networks* (LPBHN), *Transductive Classification based on Bipartite Heterogeneous Network* (TCBHN), *GNetMine* (GNM),*Tag-Based Model* (TM). Os algoritmos usados baseados no modelo espaço vetorial são: *Self-training* MNB, *Expectation Maximization* (EM) e *Transductive Singular Vector Machine* (TSVM). Os algoritmos usados baseados em rede de documentos são: *Learning With Local and Global Consistency* (LLGC), *Gaussian Fields and Harmonic Functions* (GFHF).

Os algoritmos LLGC [Zhou et al. 2003] e GFHF [Zhu et al. 2003] são utilizados para inferir a classe dos documentos utilizando uma rede de documentos baseada em similaridade. Os algoritmos TCBHN [Rossi et al. 2016], LPBHN [Zhu et al. 2003], GNetMine [Ji et al. 2010] e TagBased [Yin et al. 2009] utilizam redes bipartidas. O algoritmo *Self-Training* [Culp and Michailidis 2008, Yarowsky 1995, Haffari and Sarkar 2012] in-

duz um modelo de classificação apenas com os documentos rotulados e classifica os textos não rotulados usando esse modelo. Posteriormente, os documentos não rotulados que foram classificados com maior confinaça são inseridos no conjunto de treino. Esse processo se repete até que todos os documentos não rotulados tenham sido inseridos no conjunto de treino.

O algoritmo EM [Dempster et al. 1977] é separado em duas etapas:(i) a etapa denominada *E-step* que infere a probabilidade dos documentos; (ii) a etapa denominada *M-step* que infere a probabilidade dos termos. As etapas são executadas interativamente até a convergência, isto é, até que não haja alterações significativas nas informações de classe dos documentos não rotulados ou até um determinado número de iterações. O algoritmo TSVM [Joachims 1999] utiliza os documentos rotulados e não rotulados para induzir um hiperplano de separação máxima entre cada par de classes da coleção de textos.

3.3.4. Análise Experimental

Nesta seção são apresentados os resultados e as configurações da avaliação experimental para classificação de *tweets* utilizando algoritmos de aprendizado de máquina apresentados neste trabalho. O objetivo desta avaliação é verificar a performance de classificação dos algoritmos na classificação de sentimentos em *tweets*.

3.3.4.1. Configuração Experimental

As coleções de textos foram representadas utilizando os pré-processamentos detalhados na Seção 3.3.2. Através da base de *tweets* foram geradas três representações utilizando combinações diferentes de pré-processamento, uma representação foi radicalizada e removidas as *stopwords*, outra não foi radicalizada e foram removidas as *stopswords*, uma terceira não foi radicalizada e não foram removidas as *stopwords*. Para os algoritmos baseados no espaço-vetorial foi considerado a representação *bag-of-words*. Para algoritmos baseados em redes, foram considerados a rede bipartida e rede de documentos. Os valores das medidas de performance de classificação utilizadas no aprendizado indutivo foram $F1^{Micro}$ e $F1^{Macro}$, correspondem à media dos resultados obtidos pelo processo *10-fold cross validation* [Tan and Steinbach 2005], no aprendizado transdutivo semisupervisionado as medidas de performance de classificação utilizadas ($F1^{Micro}$ e $F1^{Macro}$) correspondem à média dessas medidas em 10 execuções. Foram utilizados algoritmos de classificação baseadas no modelos espaço-vetorial e em redes.

Os parâmetros dos algoritmos com aprendizado indutivo supervisionado são:

1. **MBN:** não há parâmetros para este algoritmo.
2. ***k*-NN:** foi utilizado o algoritmo k-NN com e sem voto ponderado pela distância. Foi considerado $k \in \{1; 3; 5; 7; 9; 11; 13; 15; 17; 19; 21; 25; 29; 35; 41; 49; 57; 73; 89\}$.
3. **SVM:** foram considerados os três tipos de kernel mais comuns, *linear*, *polynomial* (expoente = 2) a *radial basis function*, sendo que o *kernel linear* é o mais aconselhável para representações com alta dimensionalidade. Foi utilizado $C = \{10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 10^{0}, 10^{1}\}$ [Caruana and Niculescu-Mizil 2006].

4. **IMHN (IMBHN^C e IMBHN^R):** $\epsilon = 0,01$, o número máximo de iterações foi definido em 1000 e $\eta \in \{0,01; 0,05; 0,1; 0,5\}$.
5. **J48:** foram considerados os fatores de confiança como $\{0,15; 0,2; 0,25\}$

Os parâmetros dos algoritmos com aprendizado transdutivo semisupervisionado são:

1. **EM:** foi considerada a instanciação da abordagem EM apresentada em [Nigam et al. 2000] que é específica para a classificação transdutiva de textos. Foi utilizado $\lambda = \{0.1, 0.3, 0.5, 0.7, 0.9\}$ como valor de peso dado aos documentos não rotulados e $|S^{c_j}| = \{1, 2, 5, 10\}$ como número de componentes por classe.
2. **TSVM:** foi considerada a solução iterativa apresentada em [Joachims 1999] que foi utilizada para a classificação de textos. Foi utilizado $C = 1$ para induzir o hiperplano de separação considerando apenas os exemplos rotulados, uma vez que esse valor de parâmetro apresenta, em geral a melhor performance de classificação [Rossi et al. 2012]. Foi utilizado $C^{'} = \{10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 10^{0}, 10^{1}\}$ como parâmetros de punição para os documentos não rotulados que se encontram do lado incorreto do hiperplano. O algoritmo *TSVM* foi executado com e sem a função proposta em [Joachims 1999] para manter a proporção de classes dos documentos rotulados na classificação dos documentos não rotulados.
3. **LPHN:** o algoritmo LPHN não possui parâmetros a serem especificados.
4. **GFHF:** o algoritmo GFHF não possui parâmetros a serem especificados.
5. **GNetMine:** foi utilizado $\alpha = \{0.1, 0.3, 0.5, 0.7, 0.9\}$ como peso dado as informações reais durante a propagação de rótulos.
6. **TCHN:** foi utilizado $\eta = \{0,01; 0,05; 0,1; 0,5\}$, $\epsilon = 0,01$, 10 iterações máximas globais e 100 interações máximas locais, totalizando 1000 iterações.
7. **LLGC:** foi utilizado $\alpha = \{0.1, 0.3, 0.5, 0.7, 0.9\}$.
8. **TagBased:** foi utilizado $\beta = \{0.1, 1.0, 10.0, 100.0, 1000.0\}$ e $\gamma = \{0.1, 1.0, 10.0, 100.0, 1000.0\}$.
9. **Self-Training MNB:** Foi utilizado $X = 5, 10, 15, 20$ como número de exemplos não rotulados mais confiantes que são inseridos no conjunto de treinamento a cada iteração da abordagem *Self-Training*.

3.3.4.2. Resultados

Na Tabela 6 e na Tabela 7 é apresentado os resultados da classificação dos *tweets* para opinativos e não opinativos. Na Tabela 6 são apresentados os resultados dos maiores valores da medida $F1^{Micro}$ dos algoritmos de aprendizado supervisionado considerando todos os parâmetros definidos na seção anterior na rotulação de *tweets*. Na Tabela 7 são apresentados os resultados dos maiores valores da medida $F1^{Micro}$ dos algoritmos de aprendizado transdutivo semisupervisionado para cada quantidade de *tweets* rotulados. Na classificação dos *tweets* para opinativos e não opinativos o aprendizado indutivo supervisionado (KNN) no melhor caso teve uma medida $F1^{Micro}$ de 0,8114 utilizando uma base que passou pelos processos de radicalização e remoção das *stopwords*. O aprendizado transdutivo (LLGC) no melhor caso teve uma medida $F1^{Micro}$ de 0,7777 utilizando uma base sem a radicalização e sem a remoção das *stopwords*.

Na Tabela 8 e na Tabela 9 é apresentado os resultados da classificação dos *tweets* para as polaridades: positivo, negativo e neutro. Na Tabela 8 são apresentados os resultados dos maiores valores da medida $F1^{Micro}$ dos algoritmos de aprendizado supervisionado considerando todos os parâmetros definidos na seção anterior. Na Tabela 9 são apresentados os resultados dos maiores valores da medida $F1^{Micro}$ dos algoritmos de aprendizado transdutivo semisupervisionado para cada quantidade de *tweets* rotulados. Na classificação de tweets como positivos, negativos e neutros o aprendizado indutivo supervisionado (IMBHN2) teve uma medida $F1^{Micro}$ de 0,7114 utilizando uma base que passou pelos processos de radicalização e remoção das *stopwords*. O aprendizado transdutivo (LLGC) teve uma medida $F1^{Micro}$ de 0,7069 utilizando uma base que passou pelos processos de radicalização e remoção das *stopwords*.

Tabela 6. Performance de classificação dos algorítimos indutivos supervisionado na classificação de *tweets* como opinativo e não opinativo.

Algoritmo	Micro-F1	Macro-F1
KNN	0,8114	0,7816
J48	0,8057	0,7802
IMBHN	0,8047	0,7712
IMBHN2	0,7857	0,7466
MNB	0,7647	0,7478

3.4. Extração das Estatísticas Considerando os Rótulos dos Tweets

Uma vez que os resultados da análise experimental mostraram que a abordagem baseada em aprendizado de máquina utilizada neste trabalho tiveram resultados satisfatórios na classificação de *tweets*, para realizar a classificação automática foi utilizada a Text Categorization Tool (API) que implementa os algoritmos IMBHN e TCBHN [Rezende and Rossi 2016], esses algoritmos apresentaram resultados próximos aos algoritmos que tiveram melhor avaliação na Seção 3.3.4.

A API foi utilizada em duas etapas: (i) classificação automática dos *tweets* não rotulados para opinativos e não opinativos, (ii) classificação automática dos *tweets* opinativos como positivos, negativos e neutros. A quantidade de *tweets* classificados por candidato com aprendizado indutivo supervisionado são apresentados na Tabela 9. A quantidade de *tweets* classificados por candidato com aprendizado transdutivo semisupervisionado são apresentados na Tabela 10.

Tabela 7. Performance de classificação dos algorítimos trandutivos na classificação de *tweets* como opinativo e não opinativo.

	50 Exemplos Rotulados		40 Exemplos Rotulados		30 Exemplos Rotulados		20 Exemplos Rotulados		10 Exemplos Rotulados	
Algoritmo	**Micro-F1**	**Macro-F1**	**Micro-F1**	**Macro-F1**	**Micro-F1**	**Macro-F1**	**Micro-F1**	**Macro-F1**	**Micro-F1**	**Macro-F1**
LLGC	0,7777	0,7205	0,7726	0,7178	0,7703	0,7187	0,7512	0,7222	0,7466	0,7131
GFHF	0,7330	0,7075	0,7382	0,7120	0,7365	0,7130	0,7396	0,7133	0,7333	0,7116
TCBHN	0,7311	0,7248	0,7281	0,7263	0,7316	0,7231	0,7133	0,7081	0,7046	0,6953
TSVM	0,7274	0,7298	0,7268	0,7292	0,7220	0,7172	0,7111	0,7083	0,6989	0,6979
TagBased	0,7564	0,7319	0,7580	0,7419	0,7531	0,7335	0,7477	0,7221	0,7171	0,6960
LPBHN	0,7114	0,7311	0,7170	0,7347	0,7127	0,7263	0,7135	0,7217	0,7024	0,7034
GetMine	0,7361	0,7405	0,7405	0,7383	0,7342	0,7299	0,7329	0,7188	0,7187	0,7044
EM	0,7242	0,7237	0,7320	0,7170	0,7401	0,7292	0,7242	0,7200	0,7071	0,7129
SelfT MNB	0,6396	0,7087	0,6387	0,7085	0,6378	0,7071	0,6344	0,7035	0,6305	0,7002

Tabela 8. Performance de classificação dos algorítimos indutivos na classificação de *tweets* como positivos, negativos e neutros.

Algoritmo	Micro-F1	Macro-F1
IMBHN2	0,7114	0,7048
IMBHN	0,7234	0,7521
J48	0,7028	0,7095
KNN	0,6823	0,6791
MNB	0,7060	0,7029

Tabela 9. Performance de classificação dos algorítimos trandutivos na classificação de *tweets* como positivos, negativos e neutros, divididos por número de exemplos rotulados.

	50 Exemplos Rotulados		40 Exemplos Rotulados		30 Exemplos Rotulados		20 Exemplos Rotulados		10 Exemplos Rotulados	
Algoritmo	Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1
LLGC	0,7069	0,6984	0,6907	0,6839	0,6692	0,6616	0,6350	0,6267	0,6104	0,6046
GFHF	0,6839	0,6769	0,6548	0,6521	0,6388	0,6402	0,6079	0,5912	0,5967	0,5825
TCBHN	0,6839	0,6736	0,6691	0,6599	0,6659	0,6551	0,6397	0,6290	0,6156	0,6022
TSVM	0,6802	0,6707	0,6622	0,6579	0,6453	0,6459	0,6364	0,6337	0,5993	0,5965
TagBased	0,6727	0,6610	0,6682	0,6580	0,6578	0,6481	0,6263	0,6171	0,6120	0,6039
LPBHN	0,6716	0,6650	0,6612	0,6552	0,6530	0,6486	0,6274	0,6220	0,6045	0,5963
GetMine	0,6689	0,6583	0,6663	0,6571	0,6582	0,6489	0,6296	0,6205	0,6074	0,5952
EM	0,6620	0,6535	0,6456	0,6373	0,6344	0,6253	0,6122	0,6030	0,5827	0,5885
SelfT MNB	0,6251	0,6234	0,6147	0,6141	0,6026	0,6033	0,5736	0,5740	0,5325	0,5447

Para realizar a extração das estatísticas foi considerado a equação: $(pos\_candidato/tot\_tweets\_candidato) \times 100$, considerando $pos_candidato$ = número de *tweets* positivos do candidato e $tot_tweets_candidato$ = número total de *tweets* do candidato, esse calculo é utilizado para identificar o índice de aprovação do candidato. Para comparação de resultados será utilizado os dados obtidos nas pesquisas de intenção de voto obtidos pelo Ibope e DataFolha. Os resultados das pesquisas de intenção de voto são apresentadas na Tabela 12. Pare realizar o calculo de índice de rejeição foi considerado a equação $(neg_candidato/tot_tweets_candidato) \times 100$, considerando $neg_candidato$ = número de *tweets* negativos do candidato e $tot_tweets_candidato$ = número total de *tweets* do candidato. Para comparação de resultados será utilizado os dados obtidos nas pesquisas de índice de rejeição da DataFolha. Os resultados do índice de rejeição são apresentados na Tabela 13.

Utilizando os dados da Tabela 10 é possível dizer que o candidato Celso Russomanno tem 65,39% de índice de aprovação; seguido por Fernando Haddad (36,32%), Marta Suplicy (15,48%), João Doria (10,24%) e Luiza Erundina (5,93%). A liderança de Celso Russomanno se aproxima dos resultados das primeiras pesquisas do Ibope e DataFolha. Na primeira pesquisa do Ibope realizada em 21 de Junho de 2016 mostra que Celso Russomanno tem 26%; seguido por Marta Suplicy (10%), Luiza Erundina (8%), Fernando Haddad (7%) e João Doria com (6%). Na pesquisa da DataFolha de 09 de Setembro de 2016 mostra que Celso Russomanno tem 26%; seguido de Marta Suplicy (21%), João Doria (16%), Fernando Haddad (9%) e Luiza Erundia (7%). Nas pesquisas subsequentes da DataFolha de 22, 26 e 27 de Setembro de 2016, Celso Russomanno fica em segundo lugar com 22%, 24% e 22% na intenção de voto respectivamente. No resultado real da eleição tivemos João Doria com 53,29%; seguido por Fernando Haddad (16,70%), Celso Russomanno (13,64%), Marta Suplicy (10,14%) e Luiza Erundina (3,18%).

Tabela 10. Numero de tweets positivos, negativos e neutros por candidato com aprendizado Indutivo Supervisionado.

Indutivo Supervisionado			
	nº de Tweets		
Candidato	**Positivos**	**Negativos**	**Neutros**
Celso Russomanno	2491	634	684
Fernando Haddad	1751	1196	1873
João Doria	124	105	572
Luiza Erundina	89	94	686
Marta Suplicy	175	1211	1564
Total	**4630**	**3240**	**5379**

Tabela 11. Numero de tweets positivos, negativos e neutros por candidato com aprendizado Transdutivo Semisupervisionado.

Transdutivo Semisupervisionado			
	nº de Tweets		
Candidato	**Positivos**	**Negativos**	**Neutros**
Celso Russomanno	2552	752	505
Fernando Haddad	1896	1323	1601
João Doria	98	61	642
Luiza Erundina	91	54	724
Marta Suplicy	266	745	1939
Total	**4903**	**2935**	**5411**

O índice de rejeição mostra Matar Suplicy com (41,05%); seguido por Fernando Haddad (24,81%) Celso Russomanno (16,64%), João Doria (13,10%) e Luiza Erundina (10,81%). A DataFolha realizou duas pesquisas referentes ao índice de rejeição, realizadas em 09 e 22 de Setembro de 2016. Na primeira pesquisa temos Fernando Haddad com 46%; seguido por Matar Suplicy (29%), Luiza Erundina (26%), Celso Russomanno (21%) e João Doria (20%). Na segunda pesquisa temos Fernando Haddad com 45%; seguido por Matar Suplicy (29%), Luiza Erundina (27%), Celso Russomanno (27%) e João Doria (19%).

Tabela 12. Pesquisas de intenção de voto realizadas pelo Ibope e DataFolha.

Pesquisa	Ibope - 21/06/2016	DataFolha - 09/09/2017	DataFolha - 22/09/2016	DataFolha - 26/09/2016	DataFolha - 27/09/2017
Resultado	Celso Russomanno - 26%	Celso Russomanno - 26%	João Doria - 25%	João Doria - 28%	João Doria - 30%
	Marta Suplicy - 10%	Marta Suplicy - 21%	Celso Russomanno - 22%	Celso Russomanno - 24%	Celso Russomanno - 22%
	Luiza Erundina - 8%	João Doria - 16%	Marta Suplicy - 20%	Marta Suplicy - 15%	Marta Suplicy - 15%
	Fernando Haddad - 7%	Fernando Haddad - 9%	Fernando Haddad - 10%	Fernando Haddad - 12%	Fernando Haddad - 11%
	João Doria - 6%	Luiza Erundina - 7%	Luiza Erundina - 5%	Luiza Erundina - 4%	Luiza Erundina - 5%
	Outros - 43%	Outros - 21%	Outros - 18%	Outros - 17%	Outros - 18%

Comparando os resultados obtidos de índice de aprovação utilizando a Tabela 10 com os resultados reais da eleição na Tabela 14, é possivel observar que o índice de aprovação não foi capaz de acompanhar os resultados reais da eleição.

Os resultados mostram que a medição de intenção de voto e índice de rejeição por meio de análise de sentimentos não se mostrou eficiente para acompanhar os resultados das pesquisas de intenção de voto e o resultado real da eleição, porém foi capaz de

Tabela 13. Pesquisas de índice de rejeição realizadas pela DataFolha

Pesquisa	DataFolha - 26/09/2016	DataFolha - 27/09/2017
Resultado	Fernando Haddad - 45%	Fernando Haddad - 46%
	Marta Suplicy - 29%	Marta Suplicy - 29%
	Luiza Erundina - 27%	Luiza Erundina - 26%
	Celso Russomanno - 27%	Celso Russomanno - 21%
	João Doria - 19%	João Doria - 20%

Tabela 14. Resultado Real da Eleição Municial de São Paulo de 2016

Candidato - (%)
João Doria - 53,29%
Fernando Haddad - 16,70%
Celso Russomano - 13,64
Marta Suplicy - 10,14%
Luiza Erundina - 3,18%

assemelhar no posicionamento de alguns candidatos.

4. Considerações Finais

Neste artigo foram aplicadas aprendizado indutivo supervisionado e transdutivo semisupervisionado para realizar a classificação automática de *tweets*. Em geral, foram obtidos resultados satisfatórios considerados pelos autores desse trabalho para a classificação de *tweets* como opinativos e não opinativos e posteriormente como positivos, negativos e neutros.

A medição de voto por meio de aprendizado de máquina para classificação automática de textos em *tweets* não foi capaz de refletir os resultados reais da eleição, porém, obteve resultados semelhantes de algumas pesquisas de intenção de voto.

Vale ressalta que durante a construção da ferramenta para a coleta de *tweets* foi utilizado apenas a Search API. Entretanto em pesquisas posteriores foi identificado que a Streaming API é indicada para a mineração de dados nas especificações da API. Portanto, com trabalhos futuros pretende-se explorar diferentes técnicas de pré-processamento e testar diferentes algoritmos de aprendizado de máquina com intuito de obter melhores performances de classificação. Para melhorar o desempenho de classificação dos algoritmos pretende-se também aumentar o número de exemplos rotulados. Por fim, aplicar a abordagem apresentada nesse trabalho em outras eleições para uma melhor validação da abordagem utilizada neste trabalho.

Referências

Agarwal, A., Xie, B., Vovsha, I., Rambow, O., and Passonneau, R. (2011). Sentiment analysis of twitter data. In *Proceedings of the workshop on languages in social media*, pages 30–38. Association for Computational Linguistics.

Bakliwal, A., Foster, J., van der Puil, J., O'Brien, R., Tounsi, L., and Hughes, M. (2013). Sentiment analysis of political tweets: Towards an accurate classifier. Association for Computational Linguistics.

Caruana, R. and Niculescu-Mizil, A. (2006). An empirical comparison of supervised learning algorithms. In *Proceedings of the International Conference on Machine Learning*, pages 161–168. ACM.

Ceron, A., Curini, L., Iacus, S. M., and Porro, G. (2014). Every tweet counts? how sentiment analysis of social media can improve our knowledge of citizens' political preferences with an application to italy and france. *New Media & Society*, 16(2):340–358.

Culp, M. and Michailidis, G. (2008). An iterative algorithm for extending learners to a semi-supervised setting. *Journal of Computational and Graphical Statistics*, 17(3):545–571.

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society. Series B (methodological)*, pages 1–38.

Ferraz, C. et al. (1996). Crítica metodológica às pesquisas eleitorais no brasil. *Universidade Estadual de Campinas. Instituto de Matemática Estatística e Computação Científica*.

Gramacho, W. (2010). Fontes de erros das pesquisas eleitorais no brasil em 2010: uma análise exploratória. In *Anais Congresso Latino Americano de Opinião Pública. Belo Horizonte*.

Haffari, G. R. and Sarkar, A. (2012). Analysis of semi-supervised learning with the yarowsky algorithm. *arXiv preprint arXiv:1206.5240*.

Hopkins, D. J. and King, G. (2010). A method of automated nonparametric content analysis for social science. *American Journal of Political Science*, 54(1):229–247.

Ji, M., Sun, Y., Danilevsky, M., Han, J., and Gao, J. (2010). Graph regularized transductive classification on heterogeneous information networks. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 570–586. Springer.

Joachims, T. (1999). Transductive inference for text classification using support vector machines. In *Proceedings of the International Conference on Machine Learning*, pages 200–209.

Nigam, K., McCallum, A. K., Thrun, S., and Mitchell, T. (2000). Text classification from labeled and unlabeled documents using em. *Machine learning*, 39(2):103–134.

O'Connor, B., Balasubramanyan, R., Routledge, B. R., and Smith, N. A. (2010). From tweets to polls: Linking text sentiment to public opinion time series. *International Conference on Weblogs and Social Media*, 11(122-129):1–2.

REIS, A. (2003). A dança dos números: o impacto das pesquisas eleitorais nas estratégias de comunicação do HGPE na eleição de 2000 em São Paulo.

Rezende, S. O. and Rossi, R. G. (2016). "Text Categorization Tool API", instituição de registro: INPI - instituto nacional da propriedade industrial.

Rossetto, G. P. N., Carreiro, R., and Almada, M. P. (2013). Twitter e comunicação política: limites e possibilidades. *Revista Compolitica*, 3(2):189.

Rossi, R. G. (2016). *Classificação automática de textos por meio de aprendizado de máquina baseado em redes*. PhD thesis, Universidade de São Paulo.

Rossi, R. G., de Andrade Lopes, A., and Rezende, S. O. (2016). Optimization and label propagation in bipartite heterogeneous networks to improve transductive classification of texts. *Information Processing & Management*, 52(2):217–257.

Rossi, R. G., de Paulo Faleiros, T., de Andrade Lopes, A., and Rezende, S. O. (2012). Inductive model generation for text categorization using a bipartite heterogeneous network. In *Proceedings of the International Conference on Data Mining*, pages 1086–1091. IEEE.

Tan, P. and Steinbach, M. (2005). e kumar, v.(2005) Introduction to Data Mining. *Addison-Wesley Longman Publishing Co. Inc. Boston, MA, USA*.

Wang, H., Can, D., Kazemzadeh, A., Bar, F., and Narayanan, S. (2012). A system for real-time twitter sentiment analysis of 2012 us presidential election cycle. In *Proceedings of the ACL 2012 System Demonstrations*, pages 115–120. Association for Computational Linguistics.

Yarowsky, D. (1995). Unsupervised word sense disambiguation rivaling supervised methods. In *Proceedings of the 33rd annual meeting on Association for Computational Linguistics*, pages 189–196. Association for Computational Linguistics.

Yin, Z., Li, R., Mei, Q., and Han, J. (2009). Exploring social tagging graph for web object classification. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 957–966. ACM.

Zhou, D., Bousquet, O., Lal, T. N., Weston, J., and Schölkopf, B. (2003). Learning with local and global consistency. In *NIPS*, volume 16, pages 321–328.

Zhu, X., Ghahramani, Z., Lafferty, J., et al. (2003). Semi-supervised learning using gaussian fields and harmonic functions. In *ICML*, volume 3, pages 912–919.